



Sparse Approximations for Quaternionic Signals

Quentin Barthélemy, Anthony Larue, Jerome I. Mars

► To cite this version:

Quentin Barthélemy, Anthony Larue, Jerome I. Mars. Sparse Approximations for Quaternionic Signals. *Advances in Applied Clifford Algebras*, 2014, 24 (2), pp.383-402. 10.1007/s00006-013-0437-x . hal-00874959

HAL Id: hal-00874959

<https://hal.science/hal-00874959>

Submitted on 19 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sparse Approximations for Quaternionic Signals

Quentin Barthélemy, Anthony Larue and Jérôme I. Mars

Abstract. In this paper, we introduce a new processing procedure for quaternionic signals through consideration of the well-known orthogonal matching pursuit (OMP), which provides sparse approximation. Due to quaternions noncommutativity, two quaternionic extensions are presented: the right-multiplication quaternionic OMP, that can be used to process right-multiplication linear combinations of quaternionic signals, and the left-multiplication quaternionic OMP, that can be used to process left-multiplication linear combinations. As validation, these quaternionic OMP are applied to simulated data. Deconvolution is carried out and presented here with a new spikegram that is designed for visualization of quaternionic coefficients, and finally these are compared to multivariate OMP.

Keywords. Quaternionic signal processing, sparse approximation, orthogonal matching pursuit.

1. Introduction

In signal processing, some tools have been recently extended to the quaternion space \mathbb{H} . For example, we can cite quaternionic correlation for vector images [13], quaternionic SVD (SVDQ) for vector-sensor array [10], quaternionic wavelet transform (QWT) for images [5], quaternionic adaptive filtering [16], quaternionic independent component analysis [18], and blind extraction of quaternionic sources [9].

For the sparsity domain, sparse approximation algorithms [17] are given for real or complex signals. These have not been applied to quaternionic signals. Considering orthogonal matching pursuit (OMP) [14], we present two extensions to quaternions termed quaternionic OMP (Q-OMP). Due to quaternions noncommutativity, two models have to be considered: the right-multiplication model solved by Q-OMPr and the left-multiplication one solved by Q-OMPl.

In this paper, we first consider sparse approximation and the OMP algorithm in Section 2. We then present the right-multiplication and the left-multiplication linear models for quaternionic signals in Section 3. The quaternionic extensions Q-OMP_r and Q-OMP_l are detailed in Section 4. In Section 5, we specify our work for the shift-invariant case, and we introduce a new visualization tool for quaternionic sparse decompositions. Finally, in Section 6, new algorithms are applied to deconvolute simulation data, and then compared to multivariate OMP (M-OMP).

2. Sparse Approximation

The sparse approximation principle and the OMP algorithm are presented in this section, with processing of only complex signals.

2.1. Principle and existing algorithms

Considering a signal $y \in \mathbb{C}^N$ of N samples and a dictionary $\Phi \in \mathbb{C}^{N \times M}$ of M atoms $\{\phi_m\}_{m=1}^M$, the decomposition of the signal y is carried out on the dictionary Φ such that:

$$y = \Phi x + \epsilon, \quad (2.1)$$

assuming $x \in \mathbb{C}^M$ are the coding coefficients and $\epsilon \in \mathbb{C}^N$ the residual error. The dictionary is normalized, which means that its columns (atoms) are normalized, so that coefficients x reflect the energy of each atom in the signal. Moreover, the dictionary is said to be redundant when $M > N$.

One way to formalize the decomposition under the sparsity constraint is:

$$\min_x \|y - \Phi x\|_2^2 \text{ s.t. } \|x\|_0 \leq K, \quad (2.2)$$

where $K \ll M$ is a constant and $\|x\|_0$ the ℓ_0 pseudo-norm that is defined as the number of nonzero elements of vector x . This formulation is composed of a data-fitting term and a term of sparsification, to obtain the sparsest vector x . Pursuit algorithms [17] tackle sequentially (2.2) increasing K iteratively, although unfortunately this optimization is nonconvex: that means the obtained solution can get stuck in a local minimum. Nevertheless, these algorithms are fast when searching very few coefficients. Among the multiple ℓ_0 -Pursuit algorithms, we can cite the well-known matching pursuit (MP) [11], its orthogonal version, OMP [14] and multivariate OMP (M-OMP) [3] for treating multivariate signals. Note that another way consists of relaxing the sparsification term from an ℓ_0 norm to an ℓ_1 norm, which gives a convex optimization problem [17].

2.2. Review of Orthogonal Matching Pursuit

We present the step-by-step OMP that is introduced in [14] with complex signals. Given a redundant dictionary Φ , OMP produces a sparse approximation of a signal y (Algorithm 1).

After an initialization (step 1), OMP selects at the current iteration k the atom ϕ_m that produces the absolute strongest decrease in the mean

square error (MSE) $\|\epsilon^{k-1}\|_2^2$. Denoting $\epsilon^{k-1} = x_m \phi_m + \epsilon^k$, and using the complex gradient operator [4], we have:

$$\frac{\partial \|\epsilon^{k-1}\|_2^2}{\partial x_m^*} = \phi_m^H \epsilon^{k-1} = \langle \epsilon^{k-1}, \phi_m \rangle, \quad (2.3)$$

with $(.)^*$ denoting the conjugate operator and with $(.)^H$ denoting the conjugate transpose operator. Consequently, it is equivalent to selecting the atom that is the most correlated with the residue. The scalar products between the residue ϵ^{k-1} and atoms ϕ_m are computed (step 4). The selection (step 6) searches the maximum of their absolute values to determine the optimal atom ϕ_{m^k} , denoted d_k . An active dictionary $D^k \in \mathbb{C}^{N \times k}$ is formed, which collects all of the selected atoms (step 7). Coding coefficients x^k are computed via the orthogonal projection of y on D^k (step 8). This is often carried out recursively by different methods using the current correlation value $C_{m^k}^k$: QR factorization [7], Cholesky factorization [6], or block matrix inversion [14]. The obtained coefficients vector $x^k = [x_{m^1}; x_{m^2} \dots x_{m^k}]$ is reduced to its active (*i.e.* nonzero) coefficients. In the following, $[\ ; \]$ will denote the vertical concatenation, and $[\ , \]$ the horizontal one.

Different stopping criteria (step 11) can be used: a threshold on k for the number of iterations, a threshold on the relative root MSE (rRMSE) $\|\epsilon^k\|_2 / \|y\|_2$, or a threshold on the decrease in the rRMSE. At the end, the OMP provides a K -sparse approximation of y :

$$\hat{y}^K = \sum_{k=1}^K x_{m^k} \phi_{m^k}. \quad (2.4)$$

Algorithm 1 : $x = \text{OMP}(y, \Phi)$

- 1: **initialization** : $k = 1$, $\epsilon^0 = y$, dictionary $D^0 = \emptyset$
 - 2: **repeat**
 - 3: **for** $m \leftarrow 1, M$ **do**
 - 4: Scalar Products : $C_m^k \leftarrow \langle \epsilon^{k-1}, \phi_m \rangle$
 - 5: **end for**
 - 6: Selection : $m^k \leftarrow \arg \max_m |C_m^k|$
 - 7: Active Dictionary : $D^k \leftarrow [D^{k-1}, \phi_{m^k}]$
 - 8: Active Coefficients : $x^k \leftarrow \arg \min_x \|y - D^k x\|_2^2$
 - 9: Residue : $\epsilon^k \leftarrow y - D^k x^k$
 - 10: $k \leftarrow k + 1$
 - 11: **until** stopping criterion
-

Used thereafter, M-OMP [3] deals with the multivariate signals acquired simultaneously.

3. Quaternionic linear models

After a short introduction to quaternions, two linear models are investigated as a function of the variables order.

3.1. Quaternions

The quaternionic space, denoted as \mathbb{H} , is an extension of the complex space \mathbb{C} using three imaginary parts [8]. A quaternion $q \in \mathbb{H}$ is defined as:

$$q = q_a + q_b \mathbf{i} + q_c \mathbf{j} + q_d \mathbf{k}, \quad (3.1)$$

with $q_a, q_b, q_c, q_d \in \mathbb{R}$ and with the imaginary units defined as:

$$\mathbf{i}\mathbf{j} = \mathbf{k}, \mathbf{j}\mathbf{k} = \mathbf{i}, \mathbf{k}\mathbf{i} = \mathbf{j} \quad \text{and} \quad \mathbf{i}\mathbf{j}\mathbf{k} = \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1. \quad (3.2)$$

The quaternionic space is characterized by its noncommutativity: generally $q_1 q_2 \neq q_2 q_1$. The scalar part is $\mathcal{S}(q) = q_a$, and the vectorial part is $\mathcal{V}(q) = q_b \mathbf{i} + q_c \mathbf{j} + q_d \mathbf{k}$. If its scalar part is null, a quaternion is said to be pure and full otherwise. The conjugate q^* is defined as: $q^* = \mathcal{S}(q) - \mathcal{V}(q)$ and we have $(q_1 q_2)^* = q_2^* q_1^*$. The modulus is defined as: $|q| = \sqrt{q q^*}$, and the inverse is: $q^{-1} = q^* / |q|^2$.

Due to the noncommutativity of quaternions, two linear models are investigated: right-multiplication model and the left-multiplication one.

3.2. Right-multiplication linear model

The right-multiplication linear model is the natural quaternionic extension of the linear model (2.1). For the quaternionic signal $y \in \mathbb{H}^N$ of N samples and the dictionary $\Phi \in \mathbb{H}^{N \times M}$ of M atoms $\{\phi_m\}_{m=1}^M$, the decomposition of the signal y is carried out on the dictionary Φ such that:

$$y = \sum_{m=1}^M \phi_m x_m + \epsilon, \quad (3.3)$$

assuming $x_m \in \mathbb{H}$ are the coding coefficients and $\epsilon \in \mathbb{H}^N$ the residual error. This right-multiplication model is used in different real-world applications, such as wind forecasting and colored images denoising [16], and in blind source extraction of EEG mixtures [9].

Now considering quaternionic vectors $q_1, q_2 \in \mathbb{H}^N$, we define the scalar product as: $\langle q_1, q_2 \rangle = q_2^H q_1 \in \mathbb{H}$. The associated ℓ_2 norm is denoted by $\|\cdot\|_2$. Note that an alternative definition can be chosen: $\langle q_1, q_2 \rangle = q_1^H q_2$, which is only the conjugate of the previous one.¹

The sparse approximation problem for this right-multiplication model is thus the simple quaternionic extension of Eq. (2.2):

$$\min_x \|y - \Phi x\|_2^2 \quad \text{s.t.} \quad \|x\|_0 \leq K. \quad (3.4)$$

¹Note also that the scalar product in $\mathbb{R}^{N \times 4}$: $\langle q_1, q_2 \rangle = \mathcal{S}(q_1^H q_2) \in \mathbb{R}$, which is often used for quaternionic processings, is not considered here.

3.3. Left-multiplication linear model

For this left-multiplication model, the different variables have to be transposed: the studied signal is horizontal. The quaternionic signal $y \in \mathbb{H}^{1 \times N}$ of N samples, and the dictionary $\Phi \in \mathbb{H}^{M \times N}$ of M atoms $\{\phi_m\}_{m=1}^M$ are studied. The decomposition of the signal y is carried out on the dictionary Φ such that:

$$y = x \Phi + \epsilon \quad (3.5)$$

$$= \sum_{m=1}^M x_m \phi_m + \epsilon, \quad (3.6)$$

assuming $x \in \mathbb{H}^{1 \times M}$ are the coding coefficients and $\epsilon \in \mathbb{H}^{1 \times N}$ the residual error. This left-multiplication model is used for example for colored images denoising with a quaternion polynomial wavelet packet [1].

For this model, the scalar product is defined as: $\langle q_1, q_2 \rangle = q_1 q_2^H \in \mathbb{H}$, and the associated ℓ_2 norm is denoted by $\|\cdot\|_2$. The quaternionic sparse approximation problem for this left-multiplication model is written as:

$$\min_x \|y - x \Phi\|_2^2 \text{ s.t. } \|x\|_0 \leq K. \quad (3.7)$$

4. Quaternionic Orthogonal Matching Pursuits

In this section, the two algorithms solving the sparse approximation problems (3.4) and (3.7) are presented. As mentioned above, different implementations of the OMP projection step exist. In the following, we have chosen to extend the block matrix inversion method [14]. In these two cases, the dictionary is normalized, *i.e.* each atom ϕ_m is normalized. Owing to noncommutativity, the variables order is now crucial in the algorithms description.

4.1. Right-multiplication Q-OMP

The right-multiplication Q-OMP, abbreviated in Q-OMPr with r for right, solves the problem (3.4) and is described in Algorithm 2. This algorithm has been briefly introduced in [2]. As Eq. (2.3) of Algorithm 1, the scalar product $\langle \epsilon^{k-1}, \phi_m \rangle = \phi_m^H \epsilon^{k-1}$ (step 4) remains the expression to maximize to select the optimal atom (see Appendix 8.1). It is now the quaternionic scalar product defined for right-multiplication in Section 3.2.

Coefficients x^k (step 8) are calculated by orthogonal projection of y on the active dictionary $D^k \in \mathbb{H}^{N \times k}$:

$$x^k = \arg \min_x \|y - D^k x\|_2^2 \quad (4.1)$$

$$= ((D^k)^H D^k)^{-1} (D^k)^H y. \quad (4.2)$$

To compute it, the recursive procedure [14] is extended to quaternions and with the right-multiplication. Foremost, A_k is defined as the Gram matrix of

the active dictionary D^{k-1} :

$$A_k = (D^{k-1})^H D^{k-1} = \begin{bmatrix} \langle d_1, d_1 \rangle & \langle d_2, d_1 \rangle & \dots & \langle d_{k-1}, d_1 \rangle \\ \langle d_1, d_2 \rangle & \langle d_2, d_2 \rangle & \dots & \langle d_{k-1}, d_2 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle d_1, d_{k-1} \rangle & \langle d_2, d_{k-1} \rangle & \dots & \langle d_{k-1}, d_{k-1} \rangle \end{bmatrix}. \quad (4.3)$$

Remark that diagonal is equal to 1 since dictionary is normalized. At iteration k , the recursive procedure for the orthogonal projection is computed in seven stages:

- 1: $v_k = (D^{k-1})^H d_k = [\langle d_k, d_1 \rangle ; \langle d_k, d_2 \rangle \dots \langle d_k, d_{k-1} \rangle]$,
- 2: $b_k = A_k^{-1} v_k$,
- 3: $\beta = 1/(\|d_k\|_2^2 - v_k^H b_k) = 1/(1 - v_k^H b_k)$,
- 4: $\alpha_k = C_{m^k}^k \cdot \beta$.

To provide the orthogonal projection, coefficients x_{m^κ} ($\kappa = 1..k-1$) of vector x^k are corrected at each iteration. An added superscript (to coefficients x_{m^κ}) denotes the iteration, and the update is:

- 5: $x_{m^\kappa}^k = x_{m^\kappa}^{k-1} - b_k \alpha_k$, for $\kappa = 1..k-1$,
- 6: $x_{m^k}^k = \alpha_k$.

The Gram matrix update is given by:

$$A_{k+1} = \left[\begin{array}{c|c} A_k & v_k \\ \hline v_k^H & 1 \end{array} \right] \quad (4.4)$$

and using the block matrix inversion formula, we obtain its left-inverse:

$$7: \quad A_{k+1}^{-1} = \left[\begin{array}{c|c} A_k^{-1} + \beta b_k b_k^H & -\beta b_k \\ \hline -\beta b_k^H & \beta \end{array} \right]. \quad (4.5)$$

For the first iteration, the procedure is reduced to: $x_{m^1}^1 = C_{m^1}^1$ and $A_1 = 1$.

As Algorithm 1, and with the described modifications, the Q-OMP provides a K -sparse approximation of y :

$$\hat{y}^K = \sum_{k=1}^K \phi_{m^k} x_{m^k}. \quad (4.6)$$

4.2. Left-multiplication Q-OMP

The left-multiplication Q-OMP, abbreviated in Q-OMP_l with l for left, solves the problem (3.7) and is described in Algorithm 3. As Eq. (2.3), the scalar product $\langle \epsilon^{k-1}, \phi_m \rangle = \epsilon^{k-1} \phi_m^H$ (step 4) remains the expression to maximize to select the optimal atom (see Appendix 8.2). It is now the quaternionic scalar product defined for left-multiplication in Section 3.3.

Algorithm 2 : $x = \text{Q-OMPr}(y, \Phi)$

```

1: initialization :  $k = 1$ ,  $\epsilon^0 = y$ , dictionary  $D^0 = \emptyset$ 
2: repeat
3:   for  $m \leftarrow 1, M$  do
4:     Scalar Products :  $C_m^k \leftarrow \langle \epsilon^{k-1}, \phi_m \rangle = \phi_m^H \epsilon^{k-1}$ 
5:   end for
6:   Selection :  $m^k \leftarrow \arg \max_m |C_m^k|$ 
7:   Active Dictionary :  $D^k \leftarrow [D^{k-1}, \phi_{m^k}]$ 
8:   Active Coefficients :  $x^k \leftarrow \arg \min_x \|y - D^k x\|_2^2$ 
9:   Residue :  $\epsilon^k \leftarrow y - D^k x^k$ 
10:   $k \leftarrow k + 1$ 
11: until stopping criterion

```

Coefficients x^k (step 8) are calculated by orthogonal projection of y on the active dictionary $D^k \in \mathbb{H}^{k \times N}$:

$$x^k = \arg \min_x \|y - x D^k\|_2^2 \quad (4.7)$$

$$= y (D^k)^H (D^k (D^k)^H)^{-1} . \quad (4.8)$$

To compute it, the recursive procedure [14] is extended to quaternions and with the left-multiplication. Foremost, A_k is defined as the Gram matrix of the active dictionary D^{k-1} :

$$A_k = D^{k-1} (D^{k-1})^H = \begin{bmatrix} \langle d_1, d_1 \rangle & \langle d_1, d_2 \rangle & \dots & \langle d_1, d_{k-1} \rangle \\ \langle d_2, d_1 \rangle & \langle d_2, d_2 \rangle & \dots & \langle d_2, d_{k-1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle d_{k-1}, d_1 \rangle & \langle d_{k-1}, d_2 \rangle & \dots & \langle d_{k-1}, d_{k-1} \rangle \end{bmatrix} . \quad (4.9)$$

At iteration k , the recursive procedure for the orthogonal projection is computed in seven stages:

- 1: $v_k = D^{k-1} d_k^H = [\langle d_1, d_k \rangle; \langle d_2, d_k \rangle \dots \langle d_{k-1}, d_k \rangle]$,
- 2: $b_k = v_k^H A_k^{-1}$,
- 3: $\beta = 1/(1 - b_k v_k)$,
- 4: $\alpha_k = C_{m^k}^k \cdot \beta^*$.

To provide the orthogonal projection, coefficients x_{m^κ} ($\kappa = 1..k-1$) of vector x^k are corrected at each iteration. An added superscript (to coefficients x_{m^κ}) denotes the iteration, and the update is:

- 5: $x_{m^\kappa}^k = x_{m^\kappa}^{k-1} - \alpha_k b_k$, for $\kappa = 1..k-1$,
- 6: $x_{m^k}^k = \alpha_k$.

The Gram matrix update is given by:

$$A_{k+1} = \left[\begin{array}{c|c} A_k & v_k \\ \hline v_k^H & 1 \end{array} \right] \quad (4.10)$$

and using the block matrix inversion formula, we obtain its right-inverse:

$$7: \quad A_{k+1}^{-1} = \left[\begin{array}{c|c} A_k^{-1} + \beta b_k^H b_k & -\beta b_k^H \\ \hline -\beta b_k & \beta \end{array} \right]. \quad (4.11)$$

In the same way, for the first iteration, the procedure is reduced to: $x_{m^1}^1 = C_{m^1}^1$ and $A_1 = 1$.

With the described modifications, the Q-OMPl provides a K -sparse approximation of y :

$$\hat{y}^K = \sum_{k=1}^K x_{m^k} \phi_{m^k}. \quad (4.12)$$

Algorithm 3 : $x = \text{Q-OMPl}(y, \Phi)$

```

1: initialization :  $k = 1$ ,  $\epsilon^0 = y$ , dictionary  $D^0 = \emptyset$ 
2: repeat
3:   for  $m \leftarrow 1, M$  do
4:     Scalar Products :  $C_m^k \leftarrow \langle \epsilon^{k-1}, \phi_m \rangle = \epsilon^{k-1} \phi_m^H$ 
5:   end for
6:   Selection :  $m^k \leftarrow \arg \max_m |C_m^k|$ 
7:   Active Dictionary :  $D^k \leftarrow [D^{k-1}; \phi_{m^k}]$ 
8:   Active Coefficients :  $x^k \leftarrow \arg \min_x \|y - x D^k\|_2^2$ 
9:   Residue :  $\epsilon^k \leftarrow y - x^k D^k$ 
10:   $k \leftarrow k + 1$ 
11: until stopping criterion

```

5. The shift-invariant case and the spikegram

In this section, we focus on the shift-invariant case, and a new spikegram for quaternionic decompositions is introduced.

5.1. The shift-invariant case

In the shift-invariant case, we want to sparsely code the signal y as a sum of a few short structures, named kernels, that are characterized independently of their positions. This is usually applied to time series data, and this model avoids the block effects in the analysis of largely periodic signals, and provides a compact kernel dictionary [15].

The L shiftable kernels of the compact dictionary Ψ are replicated at all of the positions, to provide the M atoms of the dictionary Φ . The N samples of the signal y , the residue ϵ , and the atoms ϕ_m are indexed ² by t . The kernels $\{\psi_l\}_{l=1}^L$ can have different lengths. The kernel $\psi_l(t)$ is shifted in τ samples to generate the atom $\psi_l(t - \tau)$: zero-padding is carried out to

²Note that $a(t)$ and $a(t - t_0)$ do not represent samples, but the signal a and its translation of t_0 samples.

have N samples. The subset σ_l collects the active translations τ of the kernel $\psi_l(t)$. For the few kernels that generate all of the atoms, Eq. (2.1) becomes:

$$y(t) = \sum_{m=1}^M \phi_m(t) x_m + \epsilon(t) = \sum_{l=1}^L \sum_{\tau \in \sigma_l} \psi_l(t - \tau) x_{l,\tau} + \epsilon(t). \quad (5.1)$$

To sum up, in the shift-invariant case, the signal y is approximated as a weighted sum of a few shiftable kernels ψ_l .

5.2. Q-OMP extensions

The Q-OMP algorithms are now specified for the shift-invariant case.

For the Q-OMPr, the scalar product between the residue ϵ^{k-1} and each atom ϕ_m (step 4) is now replaced by the correlation with each kernel ψ_l . Quaternionic correlation has been introduced in [13]. In the right-multiplication case, the non-circular quaternionic correlation $\Gamma \in \mathbb{H}^{N_1+N_2-1}$ between quaternionic signals $q_1(t) \in \mathbb{H}^{N_1}$ and $q_2(t) \in \mathbb{H}^{N_2}$ is:

$$\Gamma \{q_1, q_2\}(\tau) = \langle q_1(t), q_2(t - \tau) \rangle = q_2^H(t - \tau) q_1(t). \quad (5.2)$$

The selection (step 6) determines the optimal atom that is now characterized by its kernel index l^k and its position τ^k . The orthogonal projection (step 8) gives the vector $x^k = [x_{l^1, \tau^1}; x_{l^2, \tau^2} \dots x_{l^k, \tau^k}]$. Finally, Eq. (4.6) of the K -sparse approximation becomes:

$$\hat{y}^K = \sum_{k=1}^K \psi_{l^k}(t - \tau^k) x_{l^k, \tau^k}. \quad (5.3)$$

For the Q-OMPl, the scalar product (step 4) is replaced by the quaternionic correlation defined as:

$$\Gamma \{q_1, q_2\}(\tau) = \langle q_1(t), q_2(t - \tau) \rangle = q_1(t - \tau) q_2^H(t). \quad (5.4)$$

In the same way, Equation (4.12) of the K -sparse approximation becomes:

$$\hat{y}^K = \sum_{k=1}^K x_{l^k, \tau^k} \psi_{l^k}(t - \tau^k). \quad (5.5)$$

5.3. The spikegram for quaternionic decompositions

We now explain how to visualize the coefficients obtained from a shift-invariant quaternionic decomposition. Usually, real coding coefficients $x_{l,\tau}$ are displayed by a time-kernel representation called a spikegram [15]. This condenses three indications:

- the temporal position τ (abscissa),
- the kernel index l (ordinate),
- the coefficient amplitude $x_{l,\tau}$ (gray level of the spike).

This presentation allows an intuitive readability of the decomposition. With complex coefficients, the coefficient modulus is used for the amplitude, and its argument gives the angle, which is written next to the spike [3]. This coefficient presentation provides clear visualization.

To display quaternionic coefficients and to maintain good visualization, each quaternionic coefficient is written such that:

$$x_{l,\tau} = |x_{l,\tau}| \cdot q_{l,\tau} \quad \text{and} \quad q_{l,\tau} = e^{i\theta_{l,\tau}^1} \cdot e^{k\theta_{l,\tau}^2} \cdot e^{j\theta_{l,\tau}^3}, \quad (5.6)$$

with the coefficient modulus $|x_{l,\tau}|$ that represents the atom energy, and $q_{l,\tau}$ as a unit quaternion (*i.e.* its modulus is equal to 1). This unit quaternion has only 3 degrees of freedom, which we arbitrary define as the Euler angles [8]. These parameters describe in a univocal way the considered quaternion on the unit sphere. Thereafter, we use this practical angle formalism, although without any rotation in the processing.

Two gray shading levels are set up for the quaternionic spikegram: one for coefficients amplitude, and the other for the parameters assimilated to the Euler angles. The angles scale, defined from -180 to 180 in degrees, is visually circular; a negative value just above -180 thus appears visually close to a positive value just below 180. Finally, the quaternionic coefficients $x_{l,\tau}$ are displayed in this way with six indications:

- the temporal position τ (abscissa),
- the kernel index l (ordinate),
- the coefficient amplitude $|x_{l,\tau}|$ (gray level),
- the 3 parameters $\theta_{l,\tau}^1, \theta_{l,\tau}^2, \theta_{l,\tau}^3$ displayed vertically (circular gray level).

This representation is used in the following figures and it provides an intuitive visualization of the different parameters ³.

6. Experiments on simulated data

In this section, the Q-OMPr and the Q-OMPl are illustrated on simulated signals and compared to the M-OMP.

6.1. Denoising and deconvolution

In this denoising and deconvolution experiment, the data considered are trivariate, rather than quadrivariate, only so as not to load down figures and to maintain clearer reading. Filled in pure quaternions, trivariate signals are processed using full quaternions for coding coefficients. A dictionary Ψ of $L = 6$ non-orthogonal kernels is artificially built, and five coding coefficients $x_{l,\tau}$ are generated (with overlaps between atoms). First, the quaternionic signal $y \in \mathbb{H}^N$ is formed using Eq. (5.3), which is plotted in Fig. 1(a). The first imaginary part y_b is plotted as the solid line, the second, y_c , as the dotted line, and the third, y_d , as the dashed line. Then, white Gaussian noise is added, giving the noised signal y_n that is now characterized by an SNR of 0 dB. This is shown in Fig. 1(b), maintaining the the line style convention. The generated coefficients are plotted in Fig. 1(c), using the spikegram introduced in Section 5.3.

Then, we deconvolute this signal y_n through the dictionary Ψ using Q-OMPr with $K = 5$ iterations. The denoised signal \hat{y}_n , that is obtained

³See [2] for a color version.

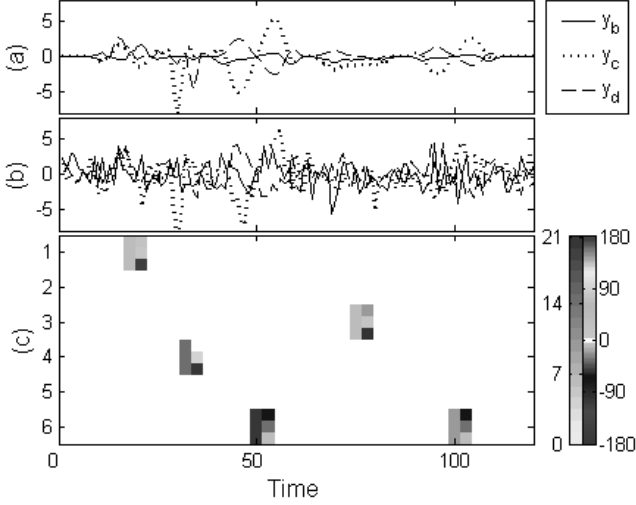


FIGURE 1. Original (a) and noised (b) quaternionic signals (first imaginary part y_b as the solid line; the second, y_c , as the dotted line; and the third, y_d , as the dashed line), and the associated spikegram (c).

by computing the K -sparse approximation of y_n , is plotted in Fig. 2(a). The coding coefficients $x_{l,\tau}$ are the result of the deconvolution, and they are shown in Fig. 2(b). Comparing Fig. 1(c) and Fig. 2(b), we observe that Q-OMPr recovers the generated coefficients well, and the approximation \hat{y}_n (Fig. 2(a)) is close to the original signal y (Fig. 1(a)); the rRMSE is 20.1 %. This experiment is randomly repeated 100 times, and the averaged rRMSE is 22.1%. This illustrates the Q-OMPr efficiency for denoising and deconvolution of quaternionic signals. In Fig. 2(b), note that coefficients $x_{6,50}$ and $x_{6,100}$ are coded with different amplitudes, but with the similar unit quaternion q .

Q-OMPI is applied to quaternionic signals with $K = 5$ iterations. The denoised signal \hat{y}_n , that is obtained by computing the K -sparse approximation of y_n , is plotted in Fig. 3(a). Comparing Fig. 1(c) and Fig. 3(b), we observe that Q-OMPI recovers the generated coefficients well, and the rRMSE is 45.8 %. This experiment is randomly repeated 100 times, and the averaged rRMSE is 35.4%.

M-OMP is now compared only using the trivariate case. The pure quaternionic signal y_n is now filled in a trivariate real signal $\underline{y}_n \in \mathbb{R}^{N \times 3}$ as well as the kernel dictionary. The M-OMP is applied with $K = 5$ iterations, and this gives the denoised signal $\hat{\underline{y}}_n$ that is plotted in Fig. 4(a). The rRMSE is 81.9%, and the average over 100 experiments is 82.7%. The associated spikegram is shown in Fig. 4(b), using the original visualization. We observe that the strong coefficients are relatively well recovered, although the others are not (temporal shift τ , kernel index l , and amplitude). However, although

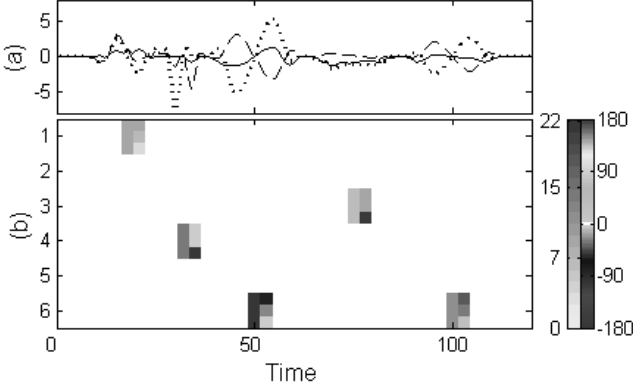


FIGURE 2. Quaternionic signal approximated by Q-OMPr (a) and the associated spikegram (b).

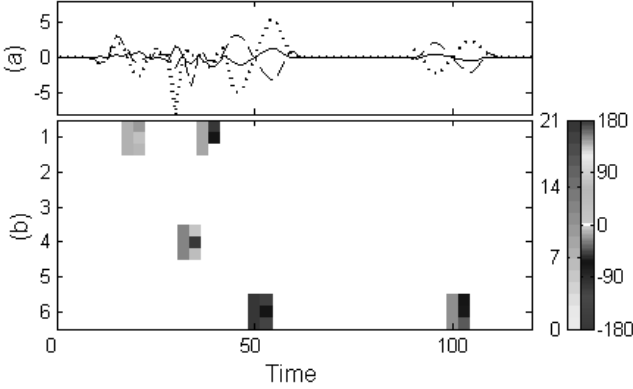


FIGURE 3. Quaternionic signal approximated by Q-OMPl (a) and the associated spikegram (b).

the strongest coefficients are recognized, this is not sufficient to obtain a satisfactory approximation. Indeed, multivariate sparse approximation is not adapted to this case, as it cannot take into account the cross-terms of the quaternionic vectorial part.

6.2. Comparisons

The three algorithms are now compared in more general conditions. 100 full quaternionic signals are studied, composed of $N = 256$ samples. As in the previous experiment, signals are generated as a linear combination of $K = 15$ atoms, with coefficients randomly drawn, but without added noise. The K -sparse approximation of signals is carried out, and the rRMSE is noted as a function of the inner iterations k of the algorithms. This error is then averaged over the 100 signals.

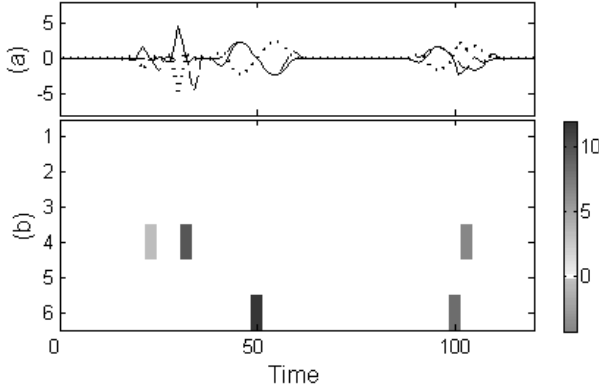


FIGURE 4. Trivariate real signal \hat{y}_n approximated by M-OMP (a) and the associated spikegram (b).

Two cases are distinguished: the first where data are generated with a right-multiplication model and results are plotted in Fig. 5; and the second with a left-multiplication model and results are plotted in Fig. 6.

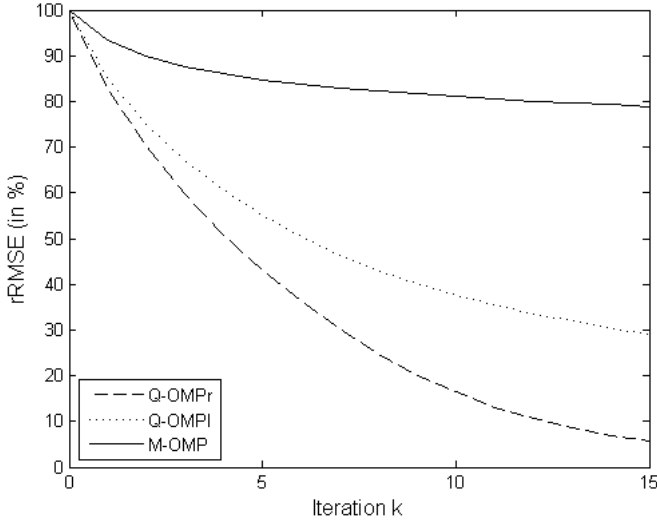


FIGURE 5. Averaged rRMSE of Q-OMPr, Q-OMPI and M-OMP on right-multiplication simulated signals, as a function of the inner iteration k .

We observe that the Q-OMPr (*resp.* Q-OMPI) gives better results for signals generated with the right-multiplication model in Fig. 5 (*resp.* the left-multiplication model in Fig. 6). It shows the necessity to have two sparse approximation algorithms. The M-OMP behaves in the same way in the two

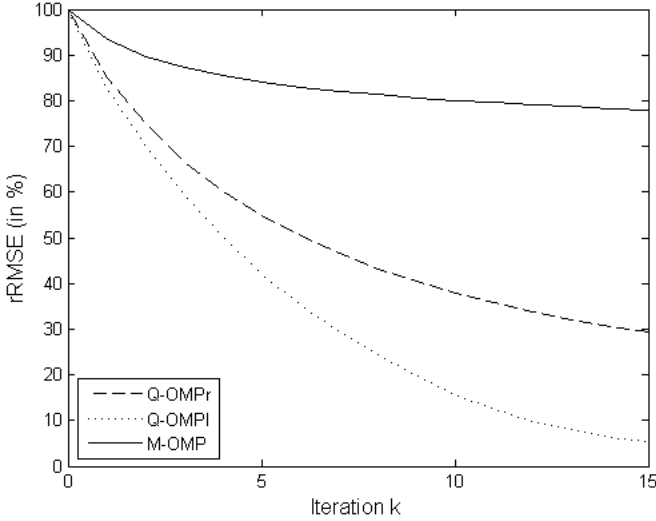


FIGURE 6. Averaged rRMSE of Q-OMPr, Q-OMPl and M-OMP on left-multiplication simulated signals, as a function of the inner iteration k .

cases, since it is not adapted for quaternionic data.

The Q-OMPr (*resp.* Q-OMPl) is not exactly at zero for $K = 15$ in Fig. 5 (*resp.* Fig. 6): this is due to the nonconvexity of the ℓ_0 -pursuit which can get stuck in a local minimum local when there are strong atoms overlaps.

6.3. Complexity

Firstly, algorithms Q-OMPr and Q-OMPl have the same complexities, since variables order only change in the different computation stages. As observed in the previous experiment, each algorithm is adapted to its multiplicative model.

A full quaternionic signal $y \in \mathbb{H}^N$ giving a quadrivariate real signal $\underline{y} \in \mathbb{R}^{N \times 4}$ is now considered. If the coefficients are strictly real, M-OMP and Q-OMPr/Q-OMPl are equivalent. If not, Q-OMPr/Q-OMPl perform better according to its multiplicative model. For the complexity, the quadrivariate correlation only has 4 terms, whereas the quaternionic one has 16: the Q-OMPr/Q-OMPl complexity is thus multiplied by 4.

7. Conclusion

We have presented here two sparse approximations algorithms for quaternionic signals. Due to the noncommutativity of quaternions, two models are considered: the Q-OMPr is dedicated to the right-multiplication linear combination of quaternionic signals, and the Q-OMPl to the left-multiplication

one. For the validation, the Q-OMPr / Q-OMPl were applied to denoise and deconvolute simulation data, and they are compared to the M-OMP.

The potential uses of Q-OMPr / Q-OMPl include quaternionic signal processing such as deconvolution, denoising, variable selection, dimensionality reduction, dictionary learning, and all of the other classical applications that are based on sparsity. Prospects are to study the extension of these algorithms to ℓ_1 -optimization algorithms [17], in order to avoid local minimum as observed in experiments.

Acknowledgment

The authors thank N. Le Bihan from GIPSA-Lab, DIS and Prof. S. Sangwine from University of Essex for their precious advises about quaternions.

8. Appendix

8.1. Selection step for Q-OMPr

For the Q-OMPr, the MSE objective function is $J = \|\epsilon\|_2^2 = \epsilon^H \epsilon$, with ϵ defined as $\epsilon = \epsilon^{k-1} = \phi x + \epsilon^k$. The derivation of J with respect to x is computed below.

To not lengthen the paper, calculus stages are not completely detailed.

$$\begin{aligned} \frac{\partial J}{\partial x} &= \frac{\partial J}{\partial x_a} + \frac{\partial J}{\partial x_b} \mathbf{i} + \frac{\partial J}{\partial x_c} \mathbf{j} + \frac{\partial J}{\partial x_d} \mathbf{k} \\ &= \frac{\partial \epsilon^H}{\partial x_a} \epsilon + \epsilon^H \frac{\partial \epsilon}{\partial x_a} + \frac{\partial \epsilon^H}{\partial x_b} \epsilon \mathbf{i} + \epsilon^H \frac{\partial \epsilon}{\partial x_b} \mathbf{i} + \frac{\partial \epsilon^H}{\partial x_c} \epsilon \mathbf{j} + \epsilon^H \frac{\partial \epsilon}{\partial x_c} \mathbf{j} + \frac{\partial \epsilon^H}{\partial x_d} \epsilon \mathbf{k} + \epsilon^H \frac{\partial \epsilon}{\partial x_d} \mathbf{k}. \end{aligned} \quad (8.1)$$

Expanding all the terms of $\epsilon = \phi x + \epsilon^k$ and $\epsilon^H = x^* \phi^H + \epsilon^{kH}$, we obtain:

$$\begin{aligned} \partial \epsilon / \partial x_a &= \phi_a + \phi_b \mathbf{i} + \phi_c \mathbf{j} + \phi_d \mathbf{k} = \phi & \partial \epsilon^H / \partial x_a &= \phi_a^T - \phi_b^T \mathbf{i} - \phi_c^T \mathbf{j} - \phi_d^T \mathbf{k} = \phi^H \\ \partial \epsilon / \partial x_b &= -\phi_b + \phi_a \mathbf{i} + \phi_d \mathbf{j} - \phi_c \mathbf{k} & \partial \epsilon^H / \partial x_b &= -\phi_b^T - \phi_a^T \mathbf{i} - \phi_d^T \mathbf{j} + \phi_c^T \mathbf{k} \\ \partial \epsilon / \partial x_c &= -\phi_c - \phi_d \mathbf{i} + \phi_a \mathbf{j} + \phi_b \mathbf{k} & \partial \epsilon^H / \partial x_c &= -\phi_c^T + \phi_d^T \mathbf{i} - \phi_a^T \mathbf{j} - \phi_b^T \mathbf{k} \\ \partial \epsilon / \partial x_d &= -\phi_d + \phi_c \mathbf{i} - \phi_b \mathbf{j} + \phi_a \mathbf{k} & \partial \epsilon^H / \partial x_d &= -\phi_d^T - \phi_c^T \mathbf{i} + \phi_b^T \mathbf{j} - \phi_a^T \mathbf{k}. \end{aligned}$$

Replacing these eight terms in Eq. (8.1), we have:

$$\begin{aligned} \frac{\partial J}{\partial x} &= \phi^H \epsilon + \epsilon^H \phi \\ &+ (-\phi_b^T - \phi_a^T \mathbf{i} - \phi_d^T \mathbf{j} + \phi_c^T \mathbf{k})(-\epsilon_b + \epsilon_a \mathbf{i} + \epsilon_d \mathbf{j} - \epsilon_c \mathbf{k}) \end{aligned} \quad (8.2)$$

$$\begin{aligned} &+ \epsilon^H (-\phi_a - \phi_b \mathbf{i} - \phi_c \mathbf{j} - \phi_d \mathbf{k}) \\ &+ (-\phi_c^T + \phi_d^T \mathbf{i} - \phi_a^T \mathbf{j} - \phi_b^T \mathbf{k})(-\epsilon_c - \epsilon_d \mathbf{i} + \epsilon_a \mathbf{j} + \epsilon_b \mathbf{k}) \end{aligned} \quad (8.3)$$

$$\begin{aligned} &+ \epsilon^H (-\phi_a - \phi_b \mathbf{i} - \phi_c \mathbf{j} - \phi_d \mathbf{k}) \\ &+ (-\phi_d^T - \phi_c^T \mathbf{i} + \phi_b^T \mathbf{j} - \phi_a^T \mathbf{k})(-\epsilon_d + \epsilon_c \mathbf{i} - \epsilon_b \mathbf{j} + \epsilon_a \mathbf{k}) \end{aligned} \quad (8.4)$$

$$\begin{aligned} &+ \epsilon^H (-\phi_a - \phi_b \mathbf{i} - \phi_c \mathbf{j} - \phi_d \mathbf{k}) \\ &= \phi^H \epsilon - 2\epsilon^H \phi + (8.2) + (8.3) + (8.4). \end{aligned} \quad (8.5)$$

Expanding the three terms (8.2), (8.3) and (8.4), adding and factorizing, we obtain:

$$(8.2) + (8.3) + (8.4) = \phi^H \epsilon + 2\epsilon^H \phi. \quad (8.6)$$

With Eq. (8.5), we finally have:

$$\begin{aligned} \frac{\partial J}{\partial x} &= \phi^H \epsilon - 2\epsilon^H \phi + \phi^H \epsilon + 2\epsilon^H \phi \\ &= 2\phi^H \epsilon = 2 \langle \epsilon, \phi \rangle. \end{aligned} \quad (8.7)$$

Thus, we can conclude that the atom which produces the strongest decrease (in absolute value) of the MSE $\|\epsilon\|_2^2$ is the most correlated to the residue, as in the complex case.

Remark that this quaternion derivation has been done with the sum of componentwise gradients. It is called pseudogradient by Mandic *et al.* who propose a quaternion gradient operator in [12]. Using their new derivative rules, we obtain $\partial J / \partial x^* = \phi^H \epsilon - 1/2 \epsilon^H \phi$. However, maximizing this expression does not give the optimal atom. It does not allow to recover known atoms in simulated signals.

8.2. Selection step for Q-OMPI

For the Q-OMPI, the MSE objective function is $J = \|\epsilon\|_2^2 = \epsilon \epsilon^H$, with ϵ defined as $\epsilon = \epsilon^{k-1} = x\phi + \epsilon^k$. The derivation of J with respect to x is computed below.

$$\begin{aligned} \frac{\partial J}{\partial x} &= \frac{\partial J}{\partial x_a} + \frac{\partial J}{\partial x_b} \mathbf{i} + \frac{\partial J}{\partial x_c} \mathbf{j} + \frac{\partial J}{\partial x_d} \mathbf{k} \\ &= \frac{\partial \epsilon}{\partial x_a} \epsilon^H + \epsilon \frac{\partial \epsilon^H}{\partial x_a} + \frac{\partial \epsilon}{\partial x_b} \epsilon^H \mathbf{i} + \epsilon \frac{\partial \epsilon^H}{\partial x_b} \mathbf{i} + \frac{\partial \epsilon}{\partial x_c} \epsilon^H \mathbf{j} + \epsilon \frac{\partial \epsilon^H}{\partial x_c} \mathbf{j} + \frac{\partial \epsilon}{\partial x_d} \epsilon^H \mathbf{k} + \epsilon \frac{\partial \epsilon^H}{\partial x_d} \mathbf{k}. \end{aligned} \quad (8.8)$$

Expanding all the terms of $\epsilon = x\phi + \epsilon^k$ and $\epsilon^H = \phi^H x^* + \epsilon^{kH}$, the eight terms of Eq. (8.8) become:

$$\begin{aligned} \frac{\partial J}{\partial x} &= \phi \epsilon^H + \epsilon \phi^H \\ &\quad + (-\phi_b + \phi_a \mathbf{i} - \phi_d \mathbf{j} + \phi_c \mathbf{k})(\epsilon_b^T + \epsilon_a^T \mathbf{i} - \epsilon_d^T \mathbf{j} + \epsilon_c^T \mathbf{k}) \end{aligned} \quad (8.9)$$

$$\begin{aligned} &\quad + \epsilon(\phi_a^T - \phi_b^T \mathbf{i} - \phi_c^T \mathbf{j} - \phi_d^T \mathbf{k}) \\ &\quad + (-\phi_c + \phi_d \mathbf{i} + \phi_a \mathbf{j} - \phi_b \mathbf{k})(\epsilon_c^T + \epsilon_d^T \mathbf{i} + \epsilon_a^T \mathbf{j} - \epsilon_b^T \mathbf{k}) \end{aligned} \quad (8.10)$$

$$\begin{aligned} &\quad + \epsilon(\phi_a^T - \phi_b^T \mathbf{i} - \phi_c^T \mathbf{j} - \phi_d^T \mathbf{k}) \\ &\quad + (-\phi_d - \phi_c \mathbf{i} + \phi_b \mathbf{j} + \phi_a \mathbf{k})(\epsilon_d^T - \epsilon_c^T \mathbf{i} + \epsilon_b^T \mathbf{j} + \epsilon_a^T \mathbf{k}) \end{aligned} \quad (8.11)$$

$$\begin{aligned} &\quad + \epsilon(\phi_a^T - \phi_b^T \mathbf{i} - \phi_c^T \mathbf{j} - \phi_d^T \mathbf{k}) \\ &= \phi \epsilon^H + \epsilon \phi^H + (8.9) + \epsilon \phi^H + (8.10) + \epsilon \phi^H + (8.11) + \epsilon \phi^H \\ &= \phi \epsilon^H + 4\epsilon \phi^H + (8.9) + (8.10) + (8.11). \end{aligned} \quad (8.12)$$

Expanding the three terms (8.9), (8.10) and (8.11), adding and factorizing, we obtain:

$$(8.9) + (8.10) + (8.11) = -\phi\epsilon^H - 2\epsilon\phi^H. \quad (8.13)$$

With Eq. (8.12), we finally have:

$$\begin{aligned} \frac{\partial J}{\partial x} &= \phi\epsilon^H + 4\epsilon\phi^H - \phi\epsilon^H - 2\epsilon\phi^H \\ &= 2\epsilon\phi^H = 2 \langle \epsilon, \phi \rangle. \end{aligned} \quad (8.14)$$

In the same way, we can conclude that the atom which produces the strongest decrease of the MSE $\|\epsilon\|_2^2$ is the most correlated to the residue. As for the Q-OMPr, the quaternion gradient operator [12], which gives $\partial J / \partial x^* = \epsilon\phi^H - 1/2 \phi\epsilon^H$, does not allow to select the optimal atom.

References

- [1] B. Augereau and P. Carré. Hypercomplex polynomial wavelets packet application for color images. In *Conf. on Applied Geometric Algebras in Computer Science and Engineering AGACSE*, 2012.
- [2] Q. Barthélemy, A. Larue, and J.I. Mars. Quaternionic sparse approximation. In *Conf. on Applied Geometric Algebras in Computer Science and Engineering AGACSE*, 2012.
- [3] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J.I. Mars. Shift & 2D rotation invariant sparse coding for multivariate signals. *IEEE Trans. on Signal Processing*, 60:1597–1611, 2012.
- [4] D.H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proceedings F - Communications, Radar and Signal Processing*, 130:11–16, 1983.
- [5] W.L. Chan, H. Choi, and R.G. Baraniuk. Coherent multiscale image processing using dual-tree quaternion wavelets. *IEEE Trans. on Image Processing*, 17:1069–1082, 2008.
- [6] S.F. Cotter, R. Adler, R.D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. *IEE Proceedings - Vision, Image and Signal Processing*, 146:235–244, 1999.
- [7] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions with matching pursuits. *Optical Engineering*, 33:2183–2191, 1994.
- [8] A.J. Hanson. *Visualizing quaternions*. Morgan-Kaufmann/Elsevier, 2006.
- [9] S. Javidi, C.C. Took, C. Jahanchahi, N. Le Bihan, and D.P. Mandic. Blind extraction of improper quaternion sources. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP '11*, pages 3708–3711, 2011.
- [10] N. Le Bihan and J. Mars. Singular value decomposition of quaternion matrices : A new tool for vector-sensor signal processing. *Signal Process.*, 84:1177–1199, 2004.
- [11] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41:3397–3415, 1993.
- [12] D.P. Mandic, C. Jahanchahi, and C.C. Took. A quaternion gradient operator and its applications. *IEEE Signal Processing Letters*, 18:47–50, 2011.

- [13] C.E. Moxey, S.J. Sangwine, and T.A. Ell. Hypercomplex correlation techniques for vector images. *IEEE Trans. on Signal Processing*, 51:1941–1953, 2003.
- [14] Y.C. Pati, R. Rezaifar, and P.S. Krishnaprasad. Orthogonal Matching Pursuit: recursive function approximation with applications to wavelet decomposition. In *Proc. Asilomar Conf. on Signals, Systems and Comput.*, pages 40–44, 1993.
- [15] E. Smith and M.S. Lewicki. Efficient coding of time-relative structure using spikes. *Neural Comput.*, 17:19–45, 2005.
- [16] C.C. Took and D.P. Mandic. Quaternion-valued stochastic gradient-based adaptive IIR filtering. *IEEE Trans. on Signal Processing*, 58:3895–3901, 2010.
- [17] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98:948–958, 2010.
- [18] J. Via, D.P. Palomar, L. Vielva, and I. Santamaria. Quaternion ICA from second-order statistics. *IEEE Trans. on Signal Processing*, 59:1586–1600, 2011.

Quentin Barthélemy

CEA, LIST

Bat DIGITEO 565 - PC 192

91191 Gif-sur-Yvette Cedex

France

e-mail: quentin.barthelemy@cea.fr

Anthony Larue

CEA, LIST

Bat DIGITEO 565 - PC 192

91191 Gif-sur-Yvette Cedex

France

e-mail: anthony.larue@cea.fr

Jérôme I. Mars

GIPSA-Lab, DIS

11 rue des mathématiques - BP 46

38402 Saint Martin d'Hères

France

e-mail: jerome.mars@gipsa-lab.grenoble-inp.fr